We claim:

Sub B1

1. A method for maintaining a data warehouse, comprising:

identifying a data source of interest;

updating a metadata to reflect information available from said source;

automatically generating a mediator based on said metadata; and

writing a wrapper for said source which calls said mediator.

2. The method of claim 1, wherein the step of updating a metadata comprises entering new types of information, new data formats for previously defined information, new transformations between data formats, and the schema of said source.

3. The method of claim 1, wherein said mediator is fully functional and is automatically generated by a stand-alone mediator generation program.

4. The method of claim 3, wherein said mediator generation program automatically defines an API and translation libraries

5. The method of claim 4, wherein said mediator comprises code to translate between source and target representations, possibly using externally defined methods, and load data into said warehouse.

6. The method of claim 1, wherein said wrapper makes use of said mediator.

7. The method of claim 3, wherein said mediator generation program defines a public data representation, wherein said wrapper uses said public data representation.

8. The method of claim 3, wherein said wrapper uses said mediator to load data into said warehouse.

9. A DataFoundry metadata model comprising abstractions, translations, mappings and database descriptions.

10. The model of claim 9, comprising a UML DataFoundry metadata representation.

11. The model of claim 9, wherein said model defines metadata used by a mediator generation program, wherein said mediator generation program generates a mediator.

12. The method of claim 11, wherein said mediator generation program comprises:

reading said metadata;

generating translation libraries;

generating an API; and

generating said mediator.

13. The method of claim 12, wherein the step of reading said metadata comprises reading the abstraction metadata; reading the translation metadata; reading the database description metadata; and reading the mapping metadata.

14. The method of claim 12, wherein the step of generating translation libraries comprises developing public and private class definitions and implementations of data structures.

15. The method of claim 14, wherein said data structures comprise said abstractions and said translations.

16. The method of claim 12, wherein generating the mediator consists of creating public and private definitions and implementations of a class or classes capable of receiving data in one format, converting it to another format, and loading it into a data warehouse.

17. The method of claim 16, wherein said data is received by a receiving data structure defined within said translation library and said data is loaded into a warehouse whose schema corresponds to the database description component of the metadata.

18. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of protein sequence and structure analysis.

19. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of functional genomics and proteomics.

20. The method of claim 1, wherein said method is used for integrating a new data source into a data warehouse.

21. The method of claim 1, wherein said method is used for updating a warehouse when a previously integrated data source is modified.

22. The model of claim 9, as defined by the UML DataFoundry representation.

23. The method of claim 14, wherein said data structures correspond to said abstractions and said translations.

24. A computer-useable medium embodying computer program code for maintaining a data warehouse by executing the steps of:

identifying a data source of interest;

updating a metadata to reflect information available from said source;

automatically generating a mediator based on said metadata; and

writing a wrapper for said source which calls said mediator.

25. The computer-useable medium of claim 24, wherein the step of updating a metadata comprises entering new types of information, new data formats for previously defined information, new transformations between data formats, and the schema of said source.

26. The computer-useable medium of claim 24, wherein said mediator is fully functional and is automatically generated by a stand-alone mediator generation program.

27. The computer-useable medium of claim 24, wherein said mediator generation program automatically defines an API and translation libraries

28. The computer-useable medium of claim 27, wherein said mediator comprises code to translate between source and target representations, possibly using externally defined methods, and load data into said warehouse.

29. The computer-useable medium of claim 24, wherein said wrapper makes use of said mediator.

30. The computer-useable medium of claim 26, wherein said mediator generation program defines a public data representation, wherein said wrapper uses said public data representation.

31. The computer-useable medium of claim 26, wherein said wrapper uses said mediator to load data into said warehouse.

32. A DataFoundry metadata model comprising abstractions, translations, mappings and database descriptions.

33. The model of claim 32, comprising a UML DataFoundry metadata representation.

34. The model of claim 32, wherein said model defines metadata used by a mediator generation program, wherein said mediator generation program generates a mediator.

35. The model of claim 34, wherein said mediator generation program comprises a computer-useable medium embodying computer program code for maintaining a data warehouse by executing the steps of:

reading said metadata;

generating translation libraries;

generating an API; and

generating said mediator.

36. The model of claim 35, wherein the step of reading said metadata comprises reading the abstraction metadata; reading the translation metadata; reading the database description metadata; and reading the mapping metadata.

37. The model of claim 35, wherein the step of generating translation libraries comprises developing public and private class definitions and implementations of data structures.

sub A2 38. The model of claim 37, wherein said data structures comprise said abstractions and said translations.

39 38. The model of claim 35, wherein generating the mediator consists of creating public and private definitions and implementations of a class or classes capable of receiving data in one format, converting it to another format, and loading it into a data warehouse.

40 39. The model of claim 38, wherein said data is received by a receiving data structure defined within said translation library and said data is loaded into a warehouse whose schema corresponds to the database description component of the metadata.

41 40. The model of claim 24, wherein said method is applied to data warehousing applications in the domain of protein sequence and structure analysis.

42 41. The model of claim 24, wherein said method is applied to data warehousing applications in the domain of functional genomics and proteomics.

43 42. The model of claim 24, wherein said method is used for integrating a new data source into a data warehouse.

44 43. The model of claim 24, wherein said method is used for updating a warehouse when a previously integrated data source is modified.

44. The model of claim 32, as defined by the UML DataFoundry representation.

45. The model of claim 37, wherein said data structures correspond to said abstractions and said translations.

46. An apparatus for maintaining a data warehouse, comprising:

means for identifying a data source of interest;

means for updating a metadata to reflect information available from said source;

means for automatically generating a mediator based on said metadata; and

means for writing a wrapper for said source which calls said mediator.

47. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of astrophysics and climate modeling.

48. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of medical image processing and analysis.

49. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of tracking consumer and customer preferences.

50. The method of claim 1, wherein said method is applied to data warehousing applications in the domain of satellite and terrestial communication systems analysis.

51. The method of claim 1, wherein said method is used for integrating a new data source into a data warehouse.

52. The method of claim 1, wherein said method is used for updating a warehouse when a previously integrated data source is modified.